**Elasticsearch**

**客户端安装**

# ElasticSearch的客户端操作

实际开发中，主要有三种方式可以作为elasticsearch服务的客户端：

第一种，elasticsearch-head插件

第二种，使用elasticsearch提供的Java的API进行访问（已经做过） --9300

第三种，使用elasticsearch提供的Restful接口直接访问（本文档研究） --9200

什么是restful？

传统方式：

URL 服务器上资源

[http://localhsot:8080/save](http://localhsot:8080/list) 访问的资源save();

[http://localhsot:8080/update](http://localhsot:8080/list) 访问的资源update();

[http://localhsot:8080/delete](http://localhsot:8080/list) 访问的资源delete();

[http://localhsot:8080/find](http://localhsot:8080/list) 访问的资源find();

Restful：表现层状态转移

HTTP的请求状态 URI 服务器上资源

Http的协议POST <http://localhsot:8080/list> 访问的资源save();

Http的协议PUT <http://localhsot:8080/list> 访问的资源update();

Http的协议DELETE http://localhsot:8080/list 访问的资源delete();

Http的协议GET http://localhsot:8080/lsit 访问的资源find();

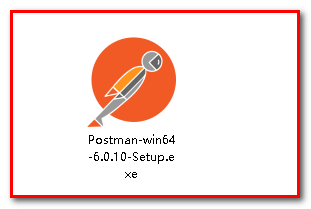
## 安装Postman工具

Postman中文版是postman这款强大网页调试工具的windows客户端，提供功能强大的Web API & HTTP 请求调试。软件功能非常强大，界面简洁明晰、操作方便快捷，设计得很人性化。Postman中文版能够发送任何类型的HTTP 请求 (GET, HEAD, POST, PUT，DELETE..)，且可以附带任何数量的参数。

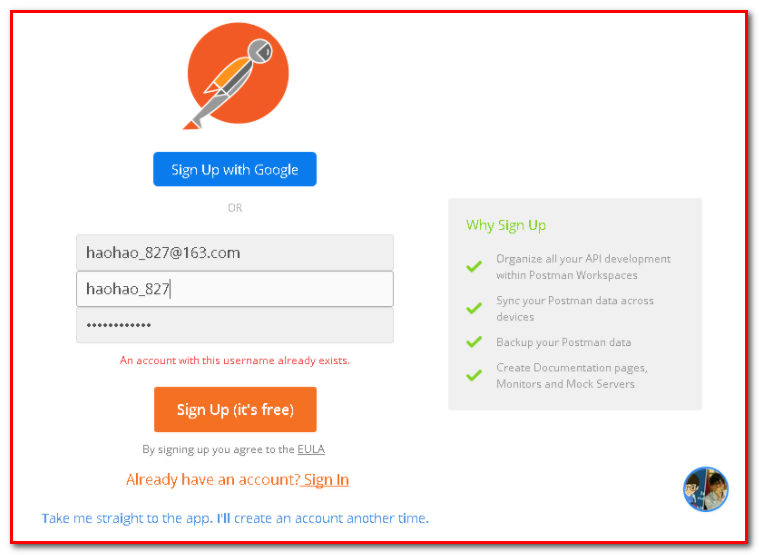
## 下载Postman工具

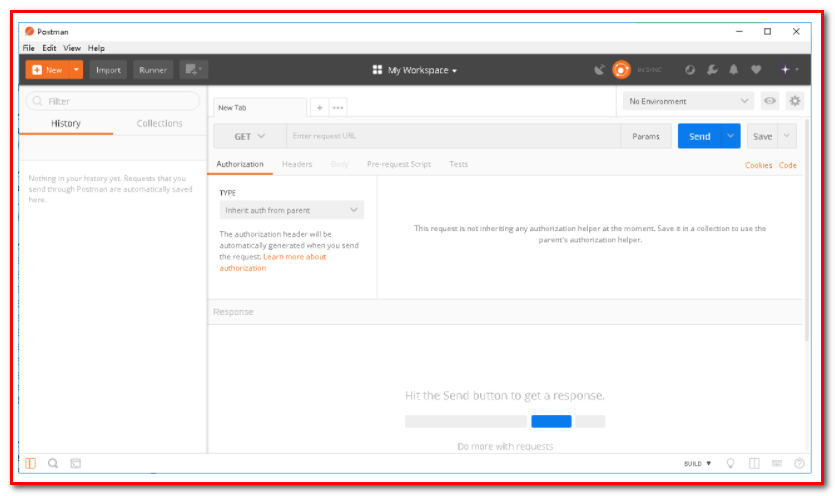
Postman官网：<https://www.getpostman.com>

课程资料中已经提供了安装包



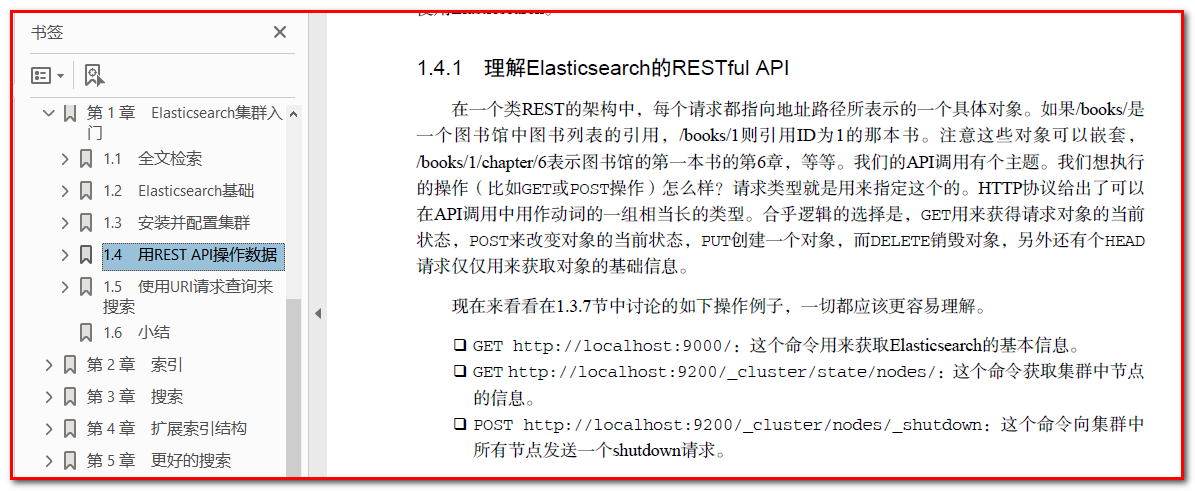
## 注册Postman工具





## 使用Postman工具进行Restful接口访问

查看文档：《Elasticsearch服务器开发（第2版）.pdf》



### ElasticSearch的接口语法

curl -X<VERB> '<PROTOCOL>://<HOST>:<PORT>/<PATH>?<QUERY\_STRING>' -d '<BODY>'

其中：

| **参数** | **解释** |
| --- | --- |
| VERB | 适当的 HTTP 方法 或 谓词 : GET、 POST、 PUT、 HEAD 或者 DELETE。 |
| PROTOCOL | http 或者 https（如果你在 Elasticsearch 前面有一个 https 代理） |
| HOST | Elasticsearch 集群中任意节点的主机名，或者用 localhost 代表本地机器上的节点。 |
| PORT | 运行 Elasticsearch HTTP 服务的端口号，默认是 9200 。 |
| PATH | API 的终端路径（例如 \_count 将返回集群中文档数量）。Path 可能包含多个组件，例如：\_cluster/stats 和 \_nodes/stats/jvm 。 |
| QUERY\_STRING | 任意可选的查询字符串参数 (例如 ?pretty 将格式化地输出 JSON 返回值，使其更容易阅读) |
| BODY | 一个 JSON 格式的请求体 (如果请求需要的话) |

### 创建索引index和映射mapping（手动）

请求url：

PUT localhost:9200/blog1

请求体：

{

"mappings": {

"article": {

"properties": {

"id": {

"type": "long",

"store": false,

"index":"not\_analyzed"

},

"title": {

"type": "text",

"store": false,

"index":"analyzed",

"analyzer":"ik\_smart"

},

"content": {

"type": "text",

"store": false,

"index":"analyzed",

"analyzer":"ik\_smart"

}

}

}

}

}

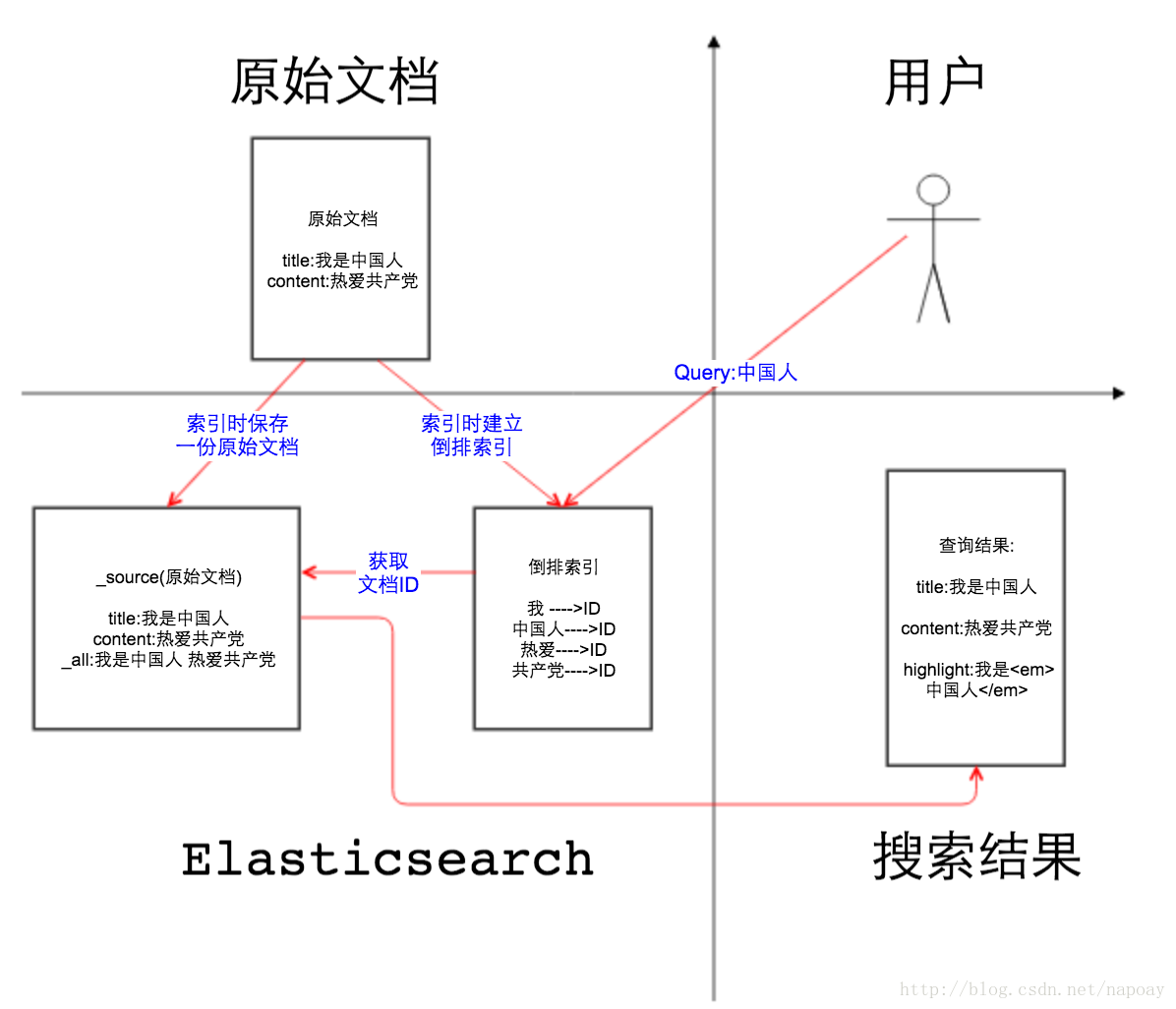
注意：

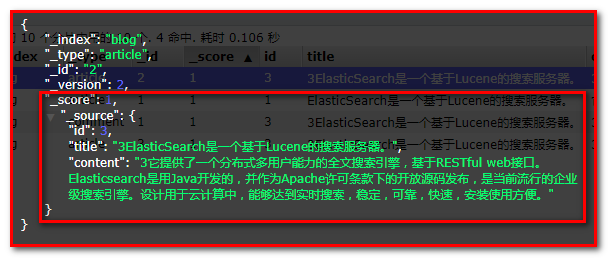
"analyzer":"standard"：表示单字分词（默认值）

"analyzer":"ik\_smart"：使用ik分词器（ik的最小切分）

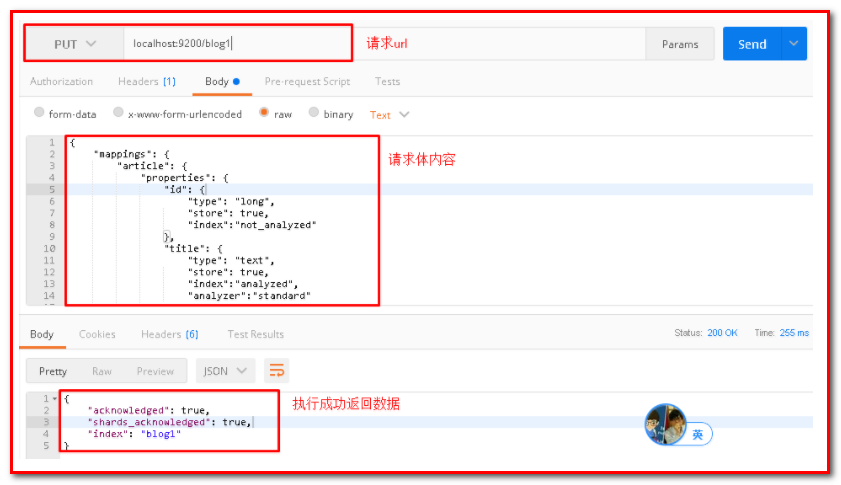
"store": true：表示是否存储，只有存储到索引库，才能检索到结果（Elasticsearch的默认值flase）

关键字高亮实质上是根据倒排记录中的词项偏移位置，找到关键词，加上前端的高亮代码。这里就要说到store属性，store属性用于指定是否将原始字段写入索引，默认取值为no。如果在Lucene中，高亮功能和store属性是否存储息息相关，因为需要根据偏移位置到原始文档中找到关键字才能加上高亮的片段。在Elasticsearch，因为\_source中已经存储了一份原始文档，可以根据\_source中的原始文档实现高亮，在索引中再存储原始文档就多余了，所以Elasticsearch默认是把store属性设置为no。

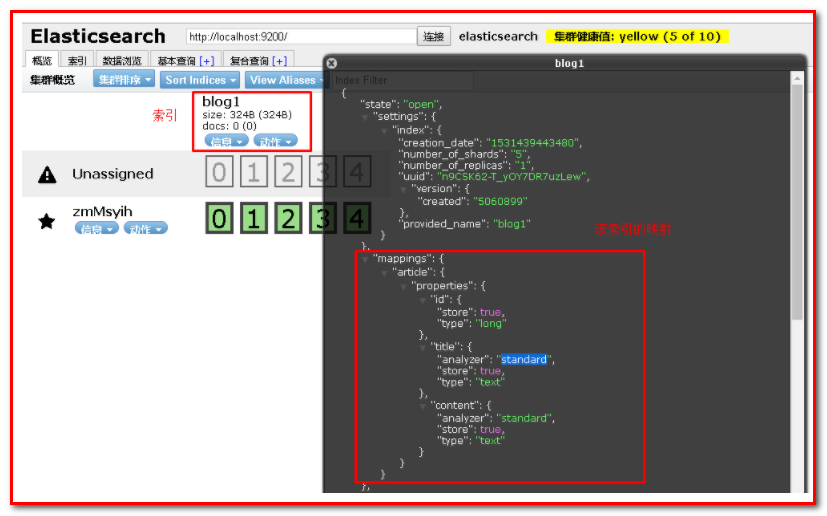




postman截图：



elasticsearch-head查看：

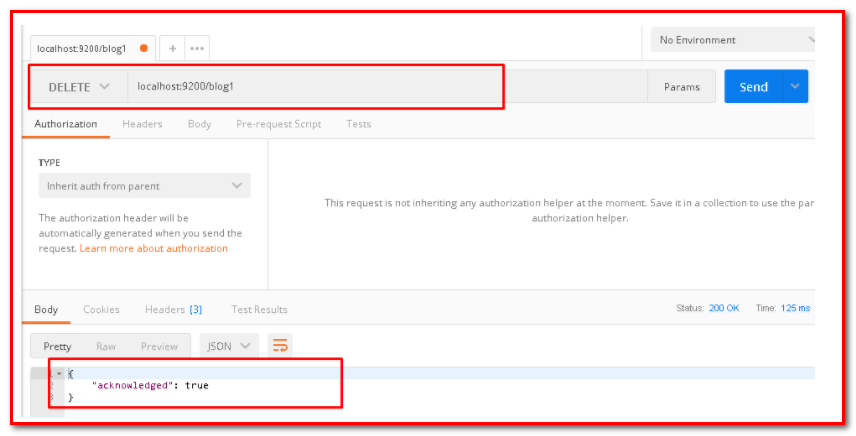


### 删除索引index

请求url：

DELETE localhost:9200/blog1

postman截图：



elasticsearch-head查看：



### 创建文档document

请求url：

POST localhost:9200/blog1/article/1

请求体：

{

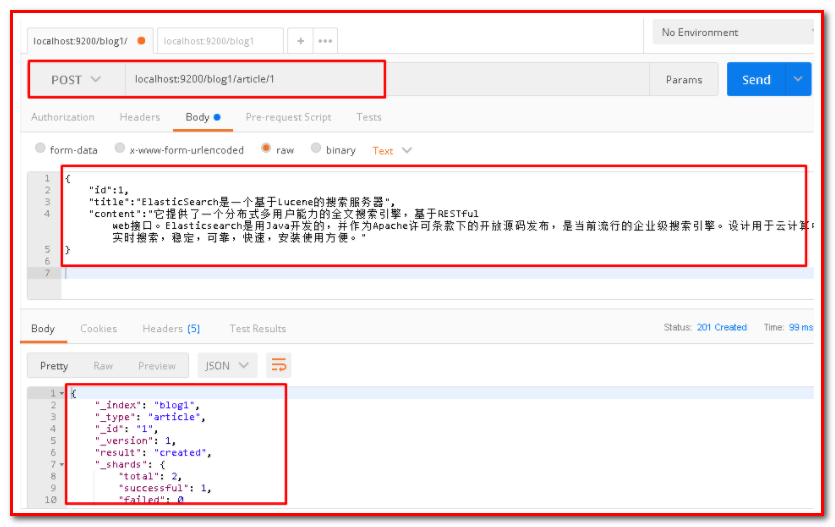
"id":1,

"title":"ElasticSearch是一个基于Lucene的搜索服务器",

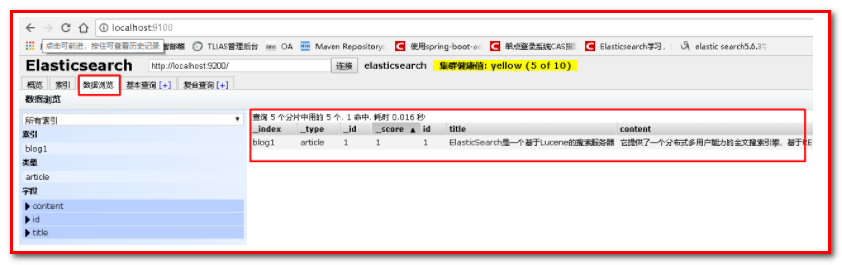
"content":"它提供了一个分布式多用户能力的全文搜索引擎，基于RESTfulweb接口。Elasticsearch是用Java开发的，并作为Apache许可条款下的开放源码发布，是当前流行的企业级搜索引擎。设计用于云计算中，能够时搜索，稳定，可靠，快速，安装使用方便。"

}

postman截图：



elasticsearch-head查看：



### 修改文档document

请求url：

POST localhost:9200/blog1/article/1

请求体：

{

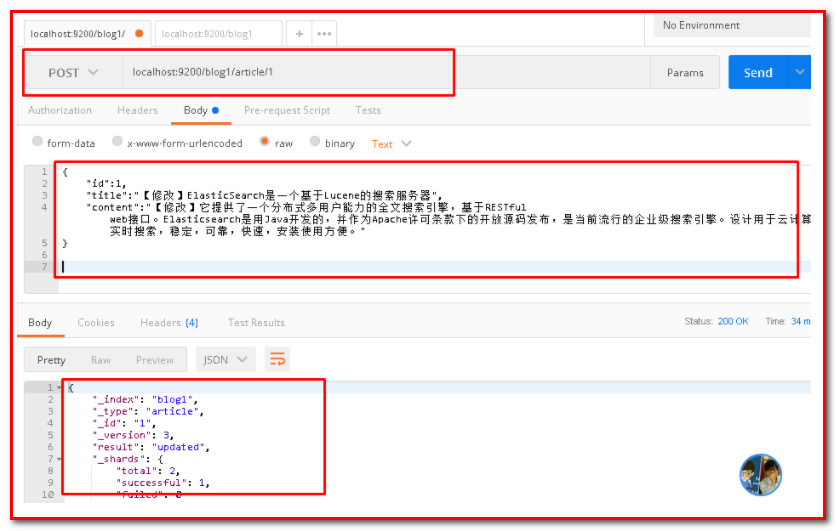
"id":1,

"title":"修改-ElasticSearch是一个基于Lucene的搜索服务器",

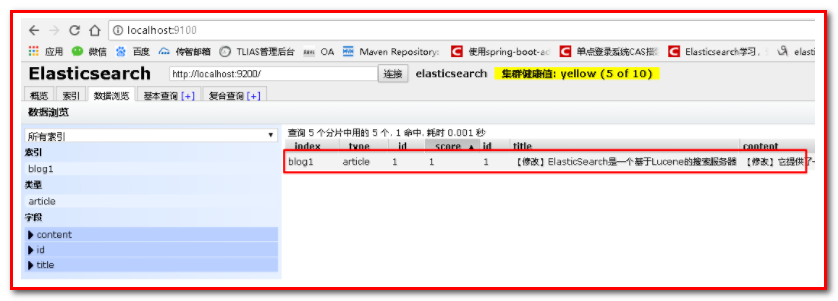
"content":"修改-它提供了一个分布式多用户能力的全文搜索引擎，基于RESTfulweb接口。Elasticsearch是用Java开发的，并作为Apache许可条款下的开放源码发布，是当前流行的企业级搜索引擎。设计用于云计算中，能够时搜索，稳定，可靠，快速，安装使用方便。"

}

postman截图：



elasticsearch-head查看：

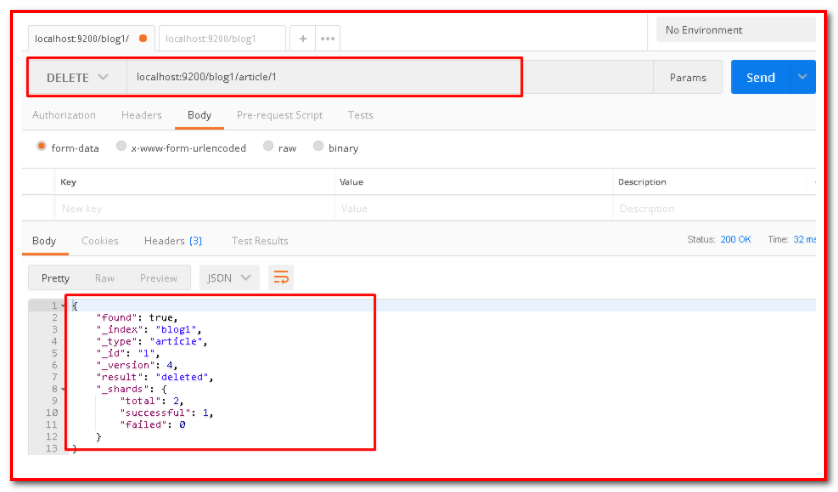


### 删除文档document

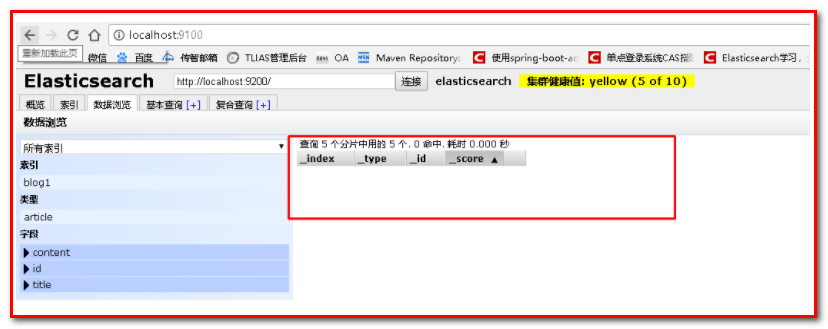
请求url：

DELETE localhost:9200/blog1/article/1

postman截图：



elasticsearch-head查看：

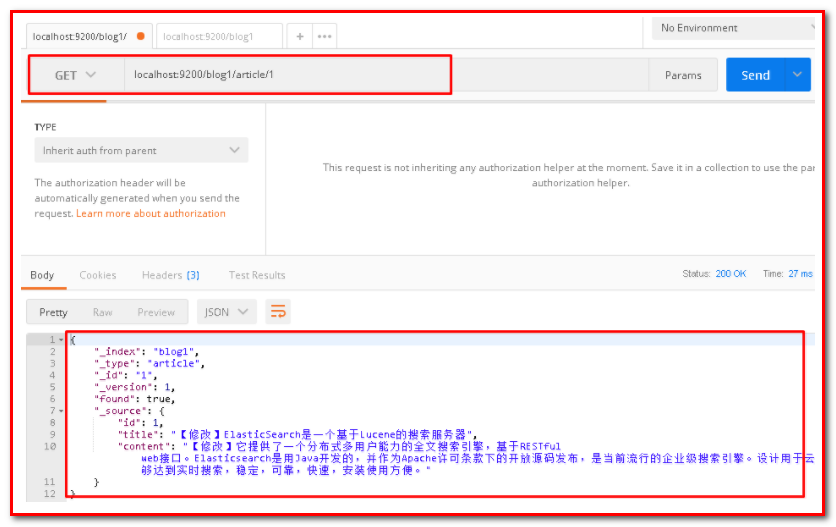


### 查询文档-根据id查询

请求url：

GET localhost:9200/blog1/article/1

postman截图：



### 查询文档-querystring查询

请求url：

POST localhost:9200/blog1/article/\_search

请求体：

{

"query": {

"query\_string": {

"default\_field": "title",

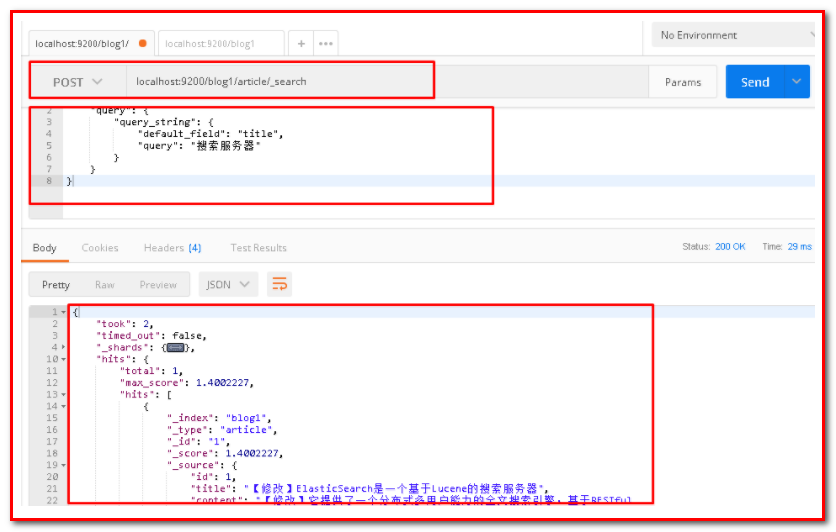
"query": "搜索服务器"

}

}

}

postman截图：



注意：

将搜索内容"搜索服务器"修改为"钢索"，同样也能搜索到文档，该原因会在下面讲解中得到答案

答案：字符串查询，先分词，再搜索、再数据合并返回。

前提：使用单字分词“standard”

{

"query": {

"query\_string": {

"default\_field": "title",

"query": "钢索服务器"

}

}

}

### 查询文档-term查询

请求url：

POST localhost:9200/blog1/article/\_search

请求体：

{

"query": {

"term": {

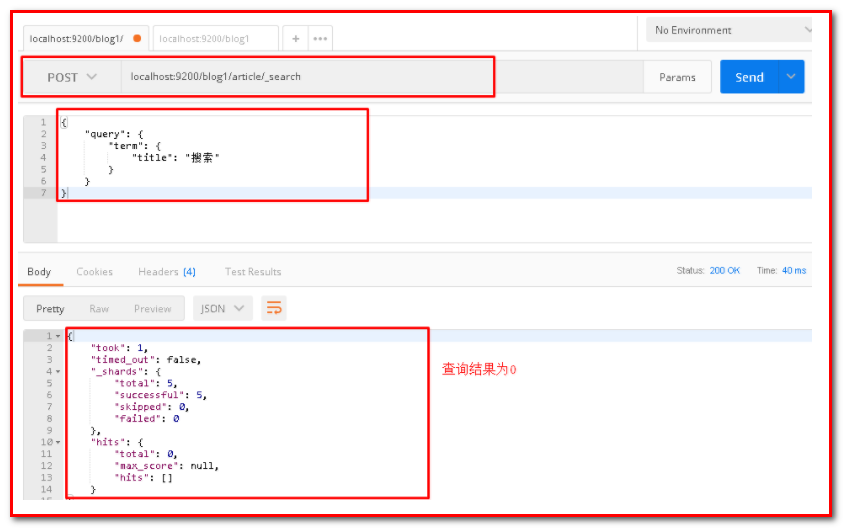
"title": "搜索"

}

}

}

postman截图：



注意：如果创建映射的时候指定ik分词器，那么就可以搜索到结果：ik\_smart表示最小切分。

{

"mappings": {

"article": {

"properties": {

"id": {

"type": "long",

"store": true,

"index":"not\_analyzed"

},

"title": {

"type": "text",

"store": true,

"index":"analyzed",

"analyzer":"ik\_smart"

},

"content": {

"type": "text",

"store": true,

"index":"analyzed",

"analyzer":"ik\_smart"

}

}

}

}

}